UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/634,304	08/04/2003	Paulo Pacheco	MWS-030RCE3	7963

74321 7590 11/10/2009

LAHIVE & COCKFIELD, LLP/THE MATHWORKS
FLOOR 30, SUITE 3000
One Post Office Square
Boston, MA 02109-2127

EXAMINER	
WANG, RONGFA PHILIP	
ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
11/10/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/634,304	PACHECO ET AL.	
	Examiner	Art Unit	
	PHILIP WANG	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 July 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5, 8-37, 39, and 42-47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5, 8-37, 39, and 42-47* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to amendment filed on 9/15/2009.

2. Per Applicant's request, claims 1, 21, 28, and 37 have been amended.

3. Claims 1-5, 8-37, 39, and 42-47 remain pending.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-3, 5, 21-23, 26-33, 36-37, 39, and 44-47 are rejected under 35 U.S.C. 103(a) as being unpatentable by Denk et al. (US PGPub. No. 2001/0025292) in view of "Digital Filter Solutions" (herein DFS).

As per claim 1,

Denk et al. disclose

A method for implementing and using a filter object the method comprising:

- providing the filter object including a state, and the filter object being represented by an output equations for generating an output of the filter object([0005], "…a

digital filter's transfer function equation…"; [0023] "…state-based…"), using the filer object in a first dynamically typed text-based programming environment ([0023], "…state-based control…"; since it is state-based, state is retained for processing; [0122], "…LMS adaptive filter…Filter 1700 includes FIR filter taps…"; see Speciation, page 6, 1st paragraph, for example, description of FIR filter; [0070], "The Matlab®…" where it shows a dynamically types text-based programming environment is used to implement such filter.)

Denk et al. do not specifically disclose

- a state equation for updating the state of the filer object implementing the filter object, and determining an output of the filter object based on an input to the filter object ,the using the filter object including:
- determining the output of the filter object including: receiving the input at the filter object,
- identifying a first state of the filter object, splitting up the input into a first input and a second input, performing a first operation by processing the output equation to determine a first output of the filter object based on the first input of the filter object and the first state of the filter object ,
- performing a second operation by processing the state equation to determine a second state of the filter object based on the input of the filter object and the first

state of the filter object, after completion of the first operation and second operation, retaining the second state of the filter object, and making the second state available after the output equation of the filter object is processed; and performing a third operation by processing the output equation to determine a second output of the filter object based on the second input of the filter object and the second state of the filter object.

However, DFS discloses

- a state equation for updating the state of the filer object implementing the filter object, and determining an output of the filter object based on an input to the filter object , (page 10, see Z transform equation, which is used to implement the filter object, where output is based on input); and using the filter object ,the using the filter object including:
- receiving an input at the filter object (page 10, see "IN" of the graphical representation showing receiving an input),
- identifying a first state of the filter object, splitting up the input into a first input and a second input, performing a first operation by processing the output equation to determine a first output of the filter object based on the first input of the filter object and the first state of the filter object (page 10, shows output "OUT' is based on input "IN" and state of the filter. The input is a continuous stream of input data, an earlier stream of data and later stream of data can be

considered as a first input and a second input. For example, the following sequence of data is being fed to the filter - (d1, d2, d3, d4) in the order from left to right. Prior to feeding d1, a first state is identified; d1 and d2 being different data, therefore, they can be split. In this case, d1 is the first input, and d2 the second input. d1 is being fed into the filter and being processed to determine a first output based on the state.),

- performing a second operation by processing the state equation to determine a second state of the filter object based on the input of the filter object and the first state of the filter object, after completion of the first operation and second operation, retaining the second state of the filter object, and making the second state available after the output equation of the filter object is processed; and performing a third operation by processing the output equation to determine a second output of the filter object based on the second input of the filter object and the second state of the filter object. (page 9, Section "Deriving Digital Filters from Z Transforms", 2nd paragraph, "...Z raised to the -1 power refers to a past value..." also see page 10, for implementation of such filter. Since past values are retained after output was processed previously, the first operation and second operation are completed and values are retained. Similarly, a

continuous stream of input data are processed based on previous input stream of data and state based data processed based on previous stream of data. Continue from the above scenario with feeding d2 and d3 into the filter.);

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of DFS into the teachings of Denk et al. to include wherein the output of the filter object is determined based on a present input of the filter object and a previous input of the filter object, and wherein the state of the filter object contains information about the previous input of the filter object . The modification would be obvious to one of ordinary skill in the art to want to be able to implement a digital filter corresponding to Z transform (page 9, Section "Deriving Digital Filters from Z Transforms", 1st paragraph).

As per claim 2,

the rejection of claim 1 is incorporated; further DFS discloses

- the filter object retains a final state obtained as after processing the input of the filter object (page 9, Section "Deriving Digital Filters from Z Transforms", 2nd paragraph, "...Z raised to the -1 power refers to a past value...").

As per claim 3,

the rejection of claim 2 is incorporated; further DFS discloses

- the final value state retained in the filter object is used as an initial state for processing a second input of the filter object(See where page 10, where more than one Z^{-1} shows past result are use for current state processing.).

As per claim 5,

the rejection of claim 1 is incorporated; further Denk et al. disclose

- the step of presetting the state of the filter object retained in the filter object([0108], line 13, "...preset the threshold value...").

As per claim 21,

- this is a computer-implemented method claim reciting similar limitations of claim 1 and is rejection for similar reason set forth in the rejection of claim 1.

As per claim 22,

the rejection of claim 21 is incorporated;

further DFS discloses

- the step of specifying equations that the system processes to generate the output of the system using the input to the system a state of the system(page 10,where input "IN" and output "OUT" is states are shown.).

As per claim 23,

the rejection of claim 21 is incorporated;

further DFS discloses

- controlling the state of a system retained in the memory (page 10, see graphical implementation, where control of state is shown).

As per claim 26,

the rejection of claim 21 is incorporated;

- It is rejected for the same reason for the rejection claim 2.

As per claim 27,

the rejection of claim 21 is incorporated;

further DFS disclose

- a state of the system is an initial state of the system for processing the input of the system (again see page 10, states of the filter is used as initial state when input is received).

As per claim 28,

- this is a computer medium claim reciting similar limitations of claim 1 and is rejection for similar reason set forth in the rejection of claim 1.
-

As per claim 29,

the rejection of claim 28 is incorporated;

further Denk et al. disclose

- the step of instantiating the object from the class ([0072]-[0076], for code example.).

As per claim 30,

the rejection of claim 28 is incorporated;

further Denk et al. disclose

- the object includes an adaptive filter object ([0028], "... adaptive filter...").

As per claim 31,

the rejection of claim 30 is incorporated;

further Denk et al. disclose

- the adaptive filter object includes an adapting algorithm that the adaptive filter implements ([0031], "...can be represented by the equation...").

As per claim 32,

the rejection of claim 28 is incorporated;

further Denk et al. disclose

- the object includes a discrete time filter object ([0034], line 9-12, "... signal sample...discrete time sequence of signal samples...").

As per claim 33,

the rejection of claim 28 is incorporated;

further Denk et al. disclose

- controlling properties of the object including a state of the object ([0108], "...preset the threshold value...selectable...").

As per claim 36,

the rejection of claim 28 is incorporated;

further Denk et al. disclose

- providing the class with methods which operate on the object of the class ([0072]-[0075]).

As per claim 37,

- it is a system claim reciting similar limitations of claim 1 and is rejected for similar reason as set forth for the rejection of claim 1.

As per claim 39,

the rejection of claim 37 is incorporated;

DFS discloses

- the new state retained in the memory is used as a state of the filter object in processing next input of the filter object(see page 10 where Z^{-1} shows a state is used for processing of next input.).

As per claim 44,

the rejection of claim 1 is incorporated; further Denk et al. disclose

- the filter object operates on a sample-by-sample, block-by-block or frame-by-frame basis([0034], line 9-12, "... signal sample...discrete time sequence of signal samples...").

As per claims 45, 46, and 47,

- they are rejected for the same reason for the rejection of claim 44.

5. Claims 4, 24, 25, 34, 42, and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Denk et al. (US PGPub. No. 2001/0025292) in view of "Digital Filter Solutions" (herein DFS) and further in view of Gay (US Patent No. 5,677,951).

As per claim 4,

the rejection of claim 1 is incorporated;

Denk et al./DFS do not specifically disclose

- the step of resetting the state of the filter object retained in the filter object.

However, Gay disclose

- the step of resetting the state of the filter object retained in the filter object (c3, line 5-26, specifically, line 11, "…a restart signal…").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Gay into the teachings Denk et al./DFS to include the step of resetting the state of the filter object retained in the filter object. The modification would be obvious to one of ordinary skill in the art to want to allow the control of the filer by a user as suggested by Gay (c3: 13-15).

As per claims 24, 34 and 42,

- they are rejected for the same reason as claim 4.

As per claim 25,

the rejection of claim 23 is incorporated;

Denk et al./DFS do not specifically disclose

- the state of the system retained in the memory is set to a particular value entered by a user.

However, Gay disclose

- the state of the system retained in the memory is set to a particular value entered by a user (c3, line 5-26, specifically, line 11, "...a restart signal...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Gay into the teachings Denk et al./DFS to include the state of the system retained in the memory is set to a particular value entered by a user. The modification would be obvious to one of ordinary skill in the art to want to allow the control of the filer by a user as suggested by Gay (c3: 13-15).

As per claim 43,

- It is rejected for the same reason as claim 25.

6. Claims 8-13 and 15-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Denk et al. (US PGPub. No. 2001/0025292) in view of "Digital Filter Solutions" (herein DFS) and further in view of "AutoCode Solutions" (herein AutoCode, http://web.archive.org/web/20021120051701/http://www.filter-solutions.com/, dated 2002).

As per claim 8,

the rejection of claim 1 is incorporated;

Denk et al./DFS do not specifically disclose

- the filter object generates code for implementing a corresponding filter algorithm.

However, Autocode discloses

- the filter object generates code for implementing a corresponding filter algorithm (page 1, 1st para., "…provides the capability to generate C code for your digital filter…").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Autocode into the teachings of Denk et al./DFS to include the filter object is utilized to generate code to implement a corresponding filter algorithm separate from the filter object implementation. The modification would be obvious to one of ordinary skill in the art to want to use the code in standard compatible compiler as suggested by Autocode (p. 1, 1st para.).

As per claim 9,

the rejection of claim 1 is incorporated;

Both Denk et al. do not disclose

- implement a corresponding test bench or filter analysis.

However, DFS discloses

- implement a corresponding test bench or filter analysis (p. 9, "Precision and Quantization", "…provide digital filter test…").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of DFS into the teachings of both Denk et al. to include "implement a corresponding test bench or filter analysis". The modification would be obvious to one of ordinary skill in the art to want to determine if a digital filter will execute

properly on the target environment as suggested by DFS (p. 9, 2nd para., line 4, "..determine if your digital filer will execute properly...).

Denk et al./DFS do not specifically disclose

- the filter object is utilized to generate code to implement a corresponding test bench or filter analysis.

However, Autocode discloses

- the filter object is utilized to generate code (page 1, 1st para., "...provides the capability to generate C code for your digital filter...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Autocode into the teachings of Denk et al./DFS to include the filter object is utilized to generate code to implement a corresponding filter algorithm separate from the filter object implementation. The modification would be obvious to one of ordinary skill in the art to want to use the code in standard compatible compiler as suggested by Autocode (p. 1, 1st para.).

As per claim 10,

the rejection of claim 8 is incorporated;

Both Denk et al.do not disclose

- the filter object execution in a simulation environment, and the generated code is executed outside a context of the simulation environment on which the filter executes;

However, DFS discloses

- the filter object execution in a simulation environment, (p. 9, "Precision and Quantization", "...the digital time response from the Filter Solution Filter control panel...provides digital filters and simulation features...on your target processor").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of DFS into the teachings of Denk et al. to include "outside the context of the simulation environment on which the filter executes". The modification would be obvious to one of ordinary skill in the art to want to determine if a digital filter will execute properly on the target environment as suggested by DFS (p. 9, 2nd para., line 4, "..determine if your digital filer will execute properly...).

Denk et al. do not disclose

- and the generated code is executed outside a context of the simulation environment on which the filter executes

However, Autocode dislcoses

- and the generated code is executed outside a context of the simulation environment on which the filter executes (p. 1, 1st para., "The code is compatible with any standard C or C++ compiler."

Where after compilation, code can be executed outside of the simulation environment.),

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Autocode into the teachings of Denk et al. to include the generated code can be executed, directly or via a suitable compilation process, on the host machine. The modification would be obvious to one of ordinary skill in the art to want to use the code in standard compatible compiler as suggested by Autocode (p. 1, 1st para.).

As per claims 11 and 12,

the rejection of claim 10 is incorporated;

further Autocode disclose

- the generated code is in a textual language/ a graphical description language (see page 2 code list in text. The examiner asserts that C can be used as a graphical description language).

As per claim 13,

the rejection of claim 8 is incorporated;

Denk et al./DFS/Autocode disclose

- the generated code executed within the context of the simulation environment in which the filter object executes (for example, DFS, page 1, first

paragraph, "…formatted for Matlab, shows the filter can be executed in a simulation environment.).

As per claim 15,

the rejection of claim 14 is incorporated;

- see reason for rejection of claim 11.

As per claim 16,

the rejection of claim 14 is incorporated;

- see reason for rejection of claim 12.

As per claim 17,

the rejection of claim 14 is incorporated;

further DFS discloses

- the generated code is suitable for use with a software implementation, the software implementation being adapted for use of the generated code at least on one of a on a general purpose processor, a digital signal processor, or other programmable compute architecture (p. 9, "Precision and Quantization", "… will execute properly on your target processor…").

As per claim 18,

the rejection of claim 14 is incorporated; further Denk et al. disclose

the generated code is suitable for use with a hardware implementation, the hardware implementation including use with t least one of a Field Programmable Gate Array (FPGA), Complex Programmable Logic Device (CPLD), and Application Specific Integrated Circuit (ASIC) device, the generate code being written in hardware description language. ([0061], "...Verilog HDL...").

As per claim 19,

the rejection of claim 8 is incorporated;

further Autocode discloses

- the code is written a high-level programming language (page 1, 1st para., "...provides the capability to generate C code for your digital filter...").

As per claim 20,

the rejection of claim 8 is incorporated; further Autocode disclose

- the code is a low-level machine or assembly language(page 1, 1st para.,"...The code is compatible with any standard C or C++ compiler".).

7. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Denk et al. (US PGPub. No. 2001/0025292) in view of "Digital Filter Solutions" (herein DFS). "AutoCode Solutions" (herein AutoCode, http://web.archive.org/web/20021120051701/http://www.filter-solutions.com/, dated 2002) and further in view of Velazquez (USPGN 2002/0121993).

As per claim 14,

the rejection of claim 8 is incorporated;

Denk et al./DFS/Autocode disclose

- the generated code is executed in an environment separate from the computer system used for a simulation of the filter object, (for example, Denk, [0061], "…Verilog HDL…"; appears to suggest it can for implementation of hardware, therefore embedded; Autocode, p. 1, 1st para., "The code is compatible with any standard C or C++ compiler." Shows generated code can be compiled to run on any machine with standard C or C++ support and such machine include machine separated from the simulation environment ;).

However, Denk et al./DFS/Autocode do not specifically disclose

- including an embedded system implementation.

However, Velazquez discloses

- including an embedded system implementation ([0091], "…cab be implemented in…firmware…").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Velazquez into the teachings of Denk et al./DFS/Autocode to include the limitation discloses by Velazquez. The modification would be obvious to one of ordinary skill in the art to want to support devices using firmware.

8. Claim 35 is rejected under 35 U.S.C. 103(a) as being unpatentable by Denk et al. (US PGPub. No. 2001/0025292) in view of "Digital Filter Solutions" (herein DFS) further in view of Pickerd (US PGPub. No. 2002/0147554).

As per claim 35,

The rejection of claim 28 is incorporated;

Denk et al/DFS do not specifically disclose

- Inheriting a state property corresponding to the state of the object from an abstract class.

However, Pickerd dislcoses

- Inheriting a state property corresponding to the state of the object from an abstract class ([0140], lines 13-15, "...a subclass object would be created. This would inherit the original abstract class behavior...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Pickerd into the teachings of Denk et al/DFS to include Inheriting a state property corresponding to the state of the object from an abstract

class. The modification would be obvious to one of ordinary skill in the art to want to minimize modification as suggested by Pickerd ([0140], line15, "...minimal...modification...").

Response to Arguments

In the remark,

1) Applicant argues:

Denk and DFS, alone or in any reasonable combination, fail to disclose or suggest "after completion of the first and second operations, retaining the second state of the filer object in the first dynamically types text-based programming environment," as recited in claim 1.

1) Examiner's response:

The Applicant remarked, "However, DFS does not disclose or suggest that the "z raised to the power -1" values are retained after completion of the processing of a previous output equation and the processing of a previous state equation, as required by amended claim 1. DFS does not disclose or suggest that, *after the Z-transform is completely processed (i. e. after an output appears at the output terminal of the Z-transform diagram),* any of the "z raised to the power -1" values in the Z-transform are retained"

In the area of Digital Signal Processing (DSP), data stream is constantly fed into the filter object as shown on page 10 of DFS. As time period runs, various sequence of data are fed into the filter object for processing. The input to "IN" is a constant stream of data. The constant stream of data can be viewed as including an earlier stream of data and a later stream of data. The earlier stream of data can be considered as the first input and the later stream can be considered the

second input. The earlier stream of data is processed and determines the state of the filter based on the equation and is later is used to process a later stream of data and in turn generate another state based on the equation. The same equation can be used to process data stream prior to the first input, and therefore the first operation and first state; the first input, and therefore the second operation and the second state; and the second input, and therefore the third operation. On page 10 of DFS, the value associated with K_{d0} can be considered as the result of performing the first operation and the value associated with K_{d1} can be considered as the result of performing the second operation. At this point the second state is retained after the completion of the first and second operations in order to process subsequent information.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Philip Wang whose telephone number is 571-272-5934. The examiner can normally be reached on Mon - Fri 8:00 - 4:00PM. Any inquiry of general nature or relating to the status of this application should be directed to the TC2100 Group receptionist: 571-272-2100.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be

obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

/Philip R. Wang/ 11/6/2009

Patent Examiner